HOW AGGRESSIVE CAN ADVERSARIAL ATTACKS BE: LEARNING ORDERED TOP-k ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep Neural Networks (DNNs) are vulnerable to adversarial attacks, especially white-box targeted attacks. This paper studies the problem of how aggressive white-box targeted attacks can be to go beyond widely used Top-1 attacks. We propose to learn **ordered Top-k attacks** ($k \geq 1$), which enforce the Top-k predicted labels of an adversarial example to be the k (randomly) selected and ordered labels (the ground-truth label is exclusive). Two methods are presented. First, we extend the vanilla Carlini-Wagner (C&W) method and use it as a strong baseline. Second, we present an **adversarial distillation** framework consisting of two components: (i) Computing an adversarial probability distribution for any given ordered Top-k targeted labels. (ii) Learning adversarial examples by minimizing the Kullback-Leibler (KL) divergence between the adversarial distribution and the predicted distribution, together with the perturbation energy penalty. In computing adversarial distributions, we explore how to leverage label semantic similarities, leading to **knowledge-oriented attacks**. In experiments, we test Top-k ($k = 1, 2, 5, 10$) attacks in the ImageNet-1000 `val` dataset using two popular DNNs trained with the clean ImageNet-1000 `train` dataset, ResNet-50 and DenseNet-121. Overall, the adversarial distillation approach obtains the best results, especially by large margin when computation budget is limited. It reduces the perturbation energy consistently with the same attack success rate on all the four k's, and improve the attack success rate by large margin against the modified C&W method for $k = 10$.

1 INTRODUCTION

Despite the recent dramatic progress, deep neural networks (DNNs) (LeCun et al., 1998; Krizhevsky et al., 2012; He et al., 2016; Szegedy et al., 2016) trained for visual recognition tasks (*e.g.*, image classification) can be easily fooled by so-called **adversarial attacks** which utilize visually imperceptible, carefully-crafted perturbations to cause networks to misclassify inputs in arbitrarily chosen ways in the close set of labels used in training (Nguyen et al., 2015; Szegedy et al., 2014; Athalye & Sutskever, 2017; Carlini & Wagner, 2016), even with one-pixel attacks (Su et al., 2017). The existence of adversarial attacks hinders the deployment of DNNs-based visual recognition systems in a wide range of applications such as autonomous driving and smart medical diagnosis in the long-run.

In this paper, we are interested in learning visually-imperceptible targeted attacks under the white-box setting in image classification tasks. In the literature, most methods address targeted attacks in the Top-1 manner, in which an adversarial attack is said to be successful if a randomly selected label (not the ground-truth label) is predicted as the Top-1 label with the added perturbation satisfying to be visually-imperceptible. One question arises,

- *The "robustness" of an attack method itself*: How far is the attack method able to push the underlying ground-truth label in the prediction of the learned adversarial examples?

Table 1 shows the evaluation results of the "robustness" of different attack methods. The widely used C&W method (Carlini & Wagner, 2016) does not push the GT labels very far, especially when smaller perturbation energy is aimed using larger search range (*e.g.*, the average rank of the GT label is 2.6 for C&W$_{9\times1000}$). Consider Top-5, if the ground-truth labels of adversarial examples still largely appear in the Top-5 of the prediction, we may be over-confident about the 100% ASR,

Table 1: Results of showing where the ground-truth (GT) labels are in the prediction of learned adversarial examples for different attack methods. The test is done in ImageNet-1000 `val` dataset using a pretrained ResNet-50 model (He et al., 2016). Please see Sec. 4 for detail of experimental settings.

Method	ASR	Proportion of GT Labels in Top-k (smaller is better)					Average Rank of GT Labels (larger is better)
		Top-3	Top-5	Top-10	Top-50	Top-100	
C&W$_{9\times30}$ (Carlini & Wagner, 2016)	99.9	36.9	50.5	66.3	90.0	95.1	20.4
C&W$_{9\times1000}$ (Carlini & Wagner, 2016)	100	71.9	87.0	96.1	99.9	100	2.6
FGSM (Goodfellow et al., 2015)	80.7	25.5	37.8	52.8	81.2	89.2	44.2
PGD$_{10}$ (Madry et al., 2018)	100	3.3	6.7	12	34.7	43.9	306.5
MIFGSM$_{10}$ (Dong et al., 2018)	99.9	0.7	1.9	6.0	22.5	32.3	404.4

especially when some downstream modules may rely on Top-5 predictions in their decision making. But, the three untargeted attack approaches are much better in terms of pushing the GT labels since they are usually move against the GT label explicitly in the optimization, but their perturbation energies are usually much larger. As we shall show, more "robust" attack methods can be developed by harnessing the advantages of the two types of attack methods. In addition, the targeted Top-1 attack setting could limit the flexibility of attacks, and may lead to less rich perturbations.

To facilitate explicit control of targeted attacks and enable more "robust" attack methods, one natural solution, which is *the focus of this paper*, is to develop **ordered Top-k targeted attacks** which enforce the Top-k predicted labels of an adversarial example to be the k (randomly) selected and ordered labels ($k \geq 1$, the GT label is exclusive). In this paper, we *present two methods* of learning ordered Top-k attacks. The basic idea is to design proper adversarial objective functions that result in imperceptible perturbations for any test image through iterative gradient-based back-propagation. *First*, we extend the vanilla Carlini-Wagner (C&W) method (Carlini & Wagner, 2016) and use it as a strong baseline. *Second*, we present an **adversarial distillation (AD)** framework consisting of two components: (i) Computing an adversarial probability distribution for any given ordered Top-k targeted labels. (ii) Learning adversarial examples by minimizing the Kullback-Leibler (KL) divergence between the adversarial distribution and the predicted distribution, together with the perturbation energy penalty.

The proposed AD framework can be viewed as applying the network distillation frameworks (Hinton et al., 2015; Bucila et al., 2006; Papernot et al., 2016) for "the bad" induced by target adversarial distributions. To compute a proper adversarial distribution for any given ordered Top-k targeted labels, the AD framework is motivated by two aspects: (i) The difference between the objective functions used by the C&W method and the three untargeted attack methods (Table 1) respectively. The former maximizes the margin of the logits between the target and the runner-up (either GT or not), while the latter maximizes the cross-entropy between the prediction probabilities (softmax of logits) and the one-hot distribution of the ground-truth. (ii) The label smoothing methods (Szegedy et al., 2015; Pereyra et al., 2017), which are often used to improve the performance of DNNs by addressing the over-confidence issue in the one-hot vector encoding of labels. More specifically, we explore how to leverage label semantic similarities in computing "smoothed" adversarial distributions, leading to **knowledge-oriented attacks**. We measure label semantic similarities using the cosine distance between some off-the-shelf word2vec embedding of labels such as the pretrained Glove embedding (Pennington et al., 2014). Along this direction, another question of interest is further investigated: *Are all Top-k targets equally challenging for an attack approach?*

In experiments, we test Top-k ($k = 1, 2, 5, 10$) in the ImageNet-1000 (Russakovsky et al., 2015) `val` dataset using two popular DNNs trained with clean ImageNet-1000 `train` dataset, ResNet-50 (He et al., 2016) and DenseNet-121 (Huang et al., 2017) respectively. Overall, the adversarial distillation approach obtains the best results. It reduces the perturbation energy consistently with the same attack success rate on all the four k's, and improve the attack success rate by large margin against the modified C&W method for $k = 10$ (see Fig. 1). We observe that Top-k targets that are distant from the GT label in terms of either label semantic distance or prediction scores of clean images are actually more difficulty to attack. In summary, *not only can ordered Top-k attacks improve the "robustness" of attacks, but also they provide insights on how aggressive adversarial attacks can be (under affordable optimization budgets).*

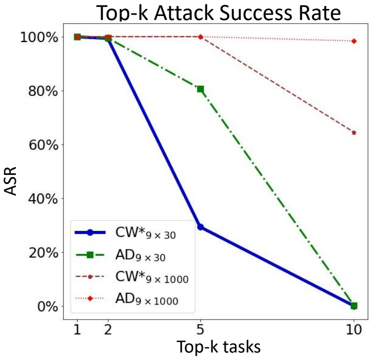

Figure 1: The average case using ResNet-50. AD is better than the modified C&W method (CW*). The thickness represents the ℓ_2 energy (thinner is better). Please see Sec. 4 for detail of experimental settings.

Our Contributions. This paper makes three main contributions to the field of learning adversarial attacks: (i) *The problem in study is novel.* Learning ordered Top-k adversarial attacks is an important problem that reflects the robustness of attacks themselves, but has not been addressed in the literature. (ii) *The proposed adversarial distillation framework is effective,* especially when k is large (such as $k = 5, 10$). (iii) The proposed knowledge-oriented adversarial distillation is novel. It worth exploring the existing distillation framework for a novel problem (ordered Top-k adversarial attacks) with some novel modifications (knowledge-oriented target distributions as "teachers").

2 Related work

The growing ubiquity of DNNs in advanced machine learning and AI systems dramatically increases their capabilities, but also increases the potential for new vulnerabilities to attacks. This situation has become critical as many powerful approaches have been developed where imperceptible perturbations to DNN inputs could deceive a well-trained DNN, significantly altering its prediction. Assuming full access to DNNs pretrained with clean images, white-box targeted attacks are powerful ways of investigating the brittleness of DNNs and their sensitivity to non-robust yet well-generalizing features in the data.

Distillation. The central idea of our proposed AD method is built on distillation. Network distillation (Bucila et al., 2006; Hinton et al., 2015) is a powerful training scheme proposed to train a new, usually lightweight model (a.k.a., the student) to mimic another already trained model (a.k.a. the teacher). It takes a functional viewpoint of the knowledge learned by the teacher as the conditional distribution it produces over outputs given an input. It teaches the student to keep up or emulate by adding some regularization terms to the loss in order to encourage the two models to be similar directly based on the distilled knowledge, replacing the training labels. Label smoothing (Szegedy et al., 2015) can be treated as a simple hand-crafted knowledge to help improve model performance. Distillation has been exploited to develop defense models (Papernot et al., 2016) to improve model robustness. Our proposed adversarial distillation method utilizes the distillation idea in an opposite direction, leveraging label semantic driven knowledge for learning ordered Top-k attacks and improving attack robustness.

Adversarial Attack. For image classification tasks using DNNs, the discovery of the existence of visually-imperceptible adversarial attacks (Szegedy et al., 2014) was a big shock in developing DNNs. White-box attacks provide a powerful way of evaluating model brittleness. In a plain and loose explanation, DNNs are universal function approximator (Hornik et al., 1989) and capable of even fitting random labels (Zhang et al., 2016) in large scale classification tasks as ImageNet-1000 (Russakovsky et al., 2015). Thus, adversarial attacks are generally learnable provided proper objective functions are given, especially when DNNs are trained with fully differentible back-propagation. Many white-box attack methods focus on norm-ball constrained objective functions (Szegedy et al., 2014; Kurakin et al., 2017; Carlini & Wagner, 2016; Dong et al., 2018). The C&W method investigates 7 different loss functions. The best performing loss function found by the C&W method has been applied in many attack methods and achieved strong results (Chen et al., 2017; Madry et al., 2018; Chen et al., 2018). By introducing momentum in the MIFGSM method (Dong et al., 2018) and the ℓ_p gradient projection in the PGD method (Madry et al., 2018), they usually achieve better performance in generating adversarial examples. In the meanwhile, some other attack methods such as the StrAttack (Xu et al., 2018) also investigate different loss functions for better interpretability of attacks. Our proposed method leverages label semantic knowledge in the loss function design for the first time.

3 Problem formulation

In this section, we first briefly introduce the white-box attack setting and the widely used C&W method (Carlini & Wagner, 2016) under the Top-1 protocol, to be self-contained. Then we define the ordered Top-k attack formulation. To learn ordered Top-k attacks, we present detail of a modified C&W method as a strong baseline and the proposed AD framework.

3.1 Background on white-box targeted attack under the Top-1 setting

We focus on classification tasks using DNNs. Denote by (x, y) a pair of a clean input $x \in \mathcal{X}$ and its ground-truth label $y \in \mathcal{Y}$. For example, in the ImageNet-1000 classification task, x represents a RGB image defined in the lattice of 224×224 and we have $\mathcal{X} \triangleq R^{3\times 224\times 224}$. y is the category label and we have $\mathcal{Y} \triangleq \{1, \cdots, 1000\}$. Let $f(\cdot; \Theta)$ be a DNN pretrained on clean training data where Θ collects all estimated parameters and is fixed in learning adversarial examples. For notation simplicity, we denote by $f(\cdot)$ a pretrained DNN. The prediction for an input x from $f(\cdot)$ is usually defined using softmax function by,

$$P = f(x) = softmax(z(x)), \tag{1}$$

where $P \in R^{|\mathcal{Y}|}$ represents the estimated confidence/probability vector ($P_c \geq 0$ and $\sum_c P_c = 1$) and $z(x)$ is the logit vector. The predicted label is then inferred by $\hat{y} = \arg\max_{c \in [1, |\mathcal{Y}|]} P_c$.

*The traditional Top-*1 *protocol of learning targeted attacks.* For an input (x, y), given a target label $t \neq y$, we seek to compute some visually-imperceptible perturbation $\delta(x, t, f)$ using the pretrained and fixed DNN $f(\cdot)$ under the white-box setting. *White-box attacks* assume the complete knowledge of the pretrained DNN f, including its parameter values, architecture, training method, etc. The perturbed example is defined by,

$$x' = x + \delta(x, t, f), \tag{2}$$

which is called **an adversarial example** of x if $t = \hat{y'} = \arg\max_c f(x')_c$ and the perturbation $\delta(x, t, f)$ is sufficiently small according to some energy metric.

The C&W Method (Carlini & Wagner, 2016). Learning $\delta(x, t, f)$ under the Top-1 protocol is posed as a constrained optimization problem (Athalye & Sutskever, 2017; Carlini & Wagner, 2016),

$$\begin{aligned} \text{minimize} \quad & \mathcal{E}(\delta) = ||\delta||_p, \\ \text{subject to} \quad & t = \arg\max_c f(x + \delta)_c, \\ & x + \delta \in [0, 1]^n, \end{aligned} \tag{3}$$

where $\mathcal{E}(\cdot)$ is defined by a ℓ_p norm (*e.g.*, the ℓ_2 norm) and n the size of the input domain (e.g., the number of pixels). To overcome the difficulty (non-linear and non-convex constraints) of directly solving Eqn. 3, the C&W method expresses it in a different form by designing some loss functions $L(x') = L(x + \delta)$ such that the first constraint $t = \arg\max_c f(x')_c$ is satisfied if and only if $L(x') \leq 0$. The best loss function proposed by the C&W method is defined by the hinge loss,

$$L_{CW}(x') = \max(0, \max_{c \neq t} z(x')_c - z(x')_t). \tag{4}$$

which induces penalties when the logit of the target label is not the maximum among all labels.

Then, the learning problem is formulated by,

$$\begin{aligned} \text{minimize} \quad & ||\delta||_p + \lambda \cdot L(x + \delta), \\ \text{subject to} \quad & x + \delta \in [0, 1]^n, \end{aligned} \tag{5}$$

which can be solved via back-propagation with the constraint satisfied via introducing a `tanh` layer. For the trade-off parameter λ, a binary search will be performed during the learning (*e.g.*, 9×1000).

3.2 THE PROPOSED ORDERED TOP-k ATTACK SETTING

It is straightforward to extend Eqn. 3 for learning ordered Top-k attacks ($k \geq 1$). Denote by $(t_1, \cdots, t_k)$ the ordered Top-k targets ($t_i \neq y$). We have,

$$\begin{aligned} \text{minimize} \quad & \mathcal{E}(\delta) = ||\delta||_p, \\ \text{subject to} \quad & t_i = \arg\max_{c \in [1, |\mathcal{Y}|], c \notin \{t_1, \cdots, t_{i-1}\}} f(x + \delta)_c, \quad i \in \{1, \cdots, k\}, \\ & x + \delta \in [0, 1]^n. \end{aligned} \tag{6}$$

Directly solving Eqn. 6 is a difficulty task and proper loss functions are entailed, similar in spirit to the approximation approaches used in the Top-1 protocol, to ensure the first constraint can be satisfied once the optimization is converged (note that the optimization may fail, *i.e.*, attacks fail).

3.3 LEARNING ORDERED TOP-k ATTACKS

3.3.1 A MODIFIED C&W METHOD

We can modify the loss function (Eqn. 4) of the C&W method accordingly to solve Eqn. 6. We have,

$$L_{CW}^{(k)}(x') = \sum_{i=1}^{k} \max\left(0, \max_{j \notin \{t_1, \cdots, t_i\}} z(x')_j - \min_{j \in \{t_1, \cdots, t_i\}} z(x')_j\right). \tag{7}$$

which covers the vanilla C&W loss (Eqn. 4), *i.e.*, when $k = 1$, $L_{CW}(x') = L_{CW}^{(1)}(x')$. The C&W loss function does not care where the underlying GT label will be as long as it is not in the Top-k.

On the one hand, it is powerful in terms of attack success rate. On the other hand, the GT label may be very close to the Top-k, leading to over-confident attacks (see Tabel. 1). In addition, it is generic for any given Top-k targets. As we will show, they are less effective if we select the Top-k targets from the sub-set of labels which are least like the ground-truth label in terms of label semantics.

3.3.2 A KNOWLEDGE-ORIENTED ADVERSARIAL DISTILLATION FRAMEWORK

To overcome the shortcomings of the C&W loss function and In our adversarial distillation framework, we adopt the view of point proposed in the network distillation method (Hinton et al., 2015) that the full confidence/probability distribution summarizes the knowledge of a trained DNN. We hypothesize that we can leverage the network distillation framework to learn the ordered Top-k attacks by designing a proper adversarial probability distribution across the entire set of labels that satisfies the specification of the given ordered Top-k targets, and facilitates explicit control of placing the GT label, as well as top-down integration of label semantics.

Consider a given set of Top-k targets, $\{t_1, \cdots, t_k\}$, denoted by P^{AD} the adversarial probability distribution in which $P^{AD}_{t_i} > P^{AD}_{t_j}$ ($\forall i < j$) and $P^{AD}_{t_i} > P^{AD}_{l}$ ($\forall l \notin \{t_1, \cdots, t_k\}$). The space of candidate distributions are huge. We present a simple knowledge-oriented approach to define the adversarial distribution. We first specify the logit distribution and then compute the probability distribution using softmax. Denote by Z the maximum logit (*e.g.*, $Z = 10$ in our experiments). We define the adversarial logits for the ordered Top-k targets by,

$$z^{AD}_{t_i} = Z - (i-1) \times \gamma, \quad i \in [1, \cdots, k], \tag{8}$$

where γ is an empirically chosen decreasing factor (*e.g.*, $\gamma = 0.3$ in our experiments). For the remaining categories $l \notin \{t_1, \cdots, t_k\}$, we define the adversarial logit by,

$$z^{AD}_{l} = \alpha \times \frac{1}{k} \sum_{i=1}^{k} s(t_i, l) + \epsilon, \tag{9}$$

where $0 \leq \alpha < z^{AD}_{t_k}$ is the maximum logit that can be assigned to any j, $s(a, b)$ is the semantic similarity between the label a and label b, and ϵ is a small position for numerical consideration (*e.g.*, $\epsilon = 1e\text{-}5$). We compute $s(a, b)$ using the cosine distance between the Glove (Pennington et al., 2014) embedding vectors of category names and $-1 \leq s(a, b) \leq 1$. Here, when $\alpha = 0$, we discard the semantic knowledge and treat all the remaining categories equally. Note that our design of P^{AD} is similar in spirit to the label smoothing technique and its variants (Szegedy et al., 2015; Pereyra et al., 2017) except that we target attack labels and exploit label semantic knowledge. The design choice is still preliminary, although we observe its effectiveness in experiments. We hope this can encourage more sophisticated work to be explored.

With the adversarial probability distribution P^{AD} defined above as the target, we use the KL divergence as the loss function in our adversarial distillation framework as done in network distillation (Hinton et al., 2015) and we have,

$$L^{(k)}_{AD}(x') = KL(f(x') || P^{AD}), \tag{10}$$

and then we follow the same optimization scheme as done in the C&W method (Eqn. 5).

4 EXPERIMENTS

In this section, we evaluate ordered Top-k attacks with $k = 1, 2, 5, 10$ in the ImageNet-1000 benchmark (Russakovsky et al., 2015) using two pretrained DNNs, ResNet-50 (He et al., 2016) and DenseNet-121 (Huang et al., 2017) from the PyTorch model zoo [1]. We implement our method using the AdverTorch toolkit [2]. Our source code will be released.

Data. In ImageNet-1000 (Russakovsky et al., 2015), there are $50,000$ images for validation. To study attacks, we utilize the subset of images for which the predictions of both the ResNet-50 and DenseNet-121 are correct. To reduce the computational demand, we randomly sample a smaller subset, as commonly done in the literature. We first randomly select 500 categories to enlarge the coverage of categories, and then randomly chose 2 images per selected categories, resulting in 1000 test images in total.

Settings. We follow the protocol used in the C&W method. We only test ℓ_2 norm as the energy penalty for perturbations in learning (Eqn. 5). But, we evaluate learned adversarial examples in

[1] https://github.com/pytorch/vision/tree/master/torchvision/models

[2] https://github.com/BorealisAI/advertorch

terms of three norms (ℓ_1, ℓ_2 and ℓ_∞). We test two search schema for the trade-off parameter λ in optimization: both use 9 steps of binary search, and 30 and 1000 iterations of optimization are performed for each trial of λ. In practice, computation budget is an important factor and less computationally expensive ones are usually preferred. Only $\alpha = 1$ is used in Eqn. 9 in experiments for simplicity due to computational demand. We compare the results under three scenarios proposed in the C&W method (Carlini & Wagner, 2016): *The Best Case* settings test the attack against all incorrect classes, and report the target class(es) that was least difficult to attack. *The Worst Case* settings test the attack against all incorrect classes, and report the target class(es) that was most difficult to attack. *The Average Case* settings select the target class(es) uniformly at random among the labels that are not the GT.

4.1 Results for ResNet-50

We first test ordered Top-k attacks using ResNet-50 for the four selected k's. Table. 2 summarizes the quantitative results and comparisons. **For Top-10 attacks**, the proposed AD method obtains significantly better results in terms of both ASR and the ℓ_2 energy of the added perturbation. For example, the proposed AD method has *relative* 362.3% *ASR improvement* over the strong C&W baseline for the worst case setting. **For Top-5 attacks**, the AD method obtains significantly better results when the search budget is relatively low (i.e., 9×30). **For Top-k ($k = 1, 2$) attacks**, both the C&W method and the AD method can achieve 100% ASR, but the AD method has consistently lower energies of the added perturbation, i.e., finding more effective attacks and richer perturbations. Fig. 2 shows some learned adversarial examples of ordered Top-10 and Top-5 attacks.

Table 2: Results and comparisons under the ordered Top-k targeted attack protocol using randomly selected and ordered 10 targets (GT exclusive) in ImageNet using ResNet-50. For Top-1 attacks, we also compare with three state-of-the-art untargeted attack methods, FGSM (Goodfellow et al., 2015), PGD (Madry et al., 2018) and MIFGSM (Dong et al., 2018). 10 iterations are used for both PGD and MIFGSM.

Protocol	Attack Method	Best Case				Average Case				Worst Case			
		ASR	ℓ_1	ℓ_2	ℓ_∞	ASR	ℓ_1	ℓ_2	ℓ_∞	ASR	ℓ_1	ℓ_2	ℓ_∞
Top-10	C&W$^*_{9\times30}$	0	N.A.	N.A.	N.A.	0	N.A.	N.A.	N.A.	0	N.A.	N.A.	N.A.
	AD$_{9\times30}$	0.8	2579	8.18	0.096	0.16	2579	8.18	0.096	0	N.A.	N.A.	N.A.
	C&W$^*_{9\times100}$	43.4	2336	7.83	0.109	11.8	2330	7.82	0.109	0.1	2479	8.26	0.119
	AD$_{9\times100}$	**91.8**	1677	5.56	0.088	**51.2**	1867	6.14	0.098	**5.6**	2021	6.62	0.110
	C&W$^*_{9\times1000}$	97.7	1525	5.26	0.092	64.5	1742	5.99	0.103	20.4	1898	6.61	0.120
	AD$_{9\times1000}$	**99.8**	**678**	**2.45**	**0.060**	**98.4**	**974**	**3.45**	**0.081**	**94.3**	**1278**	**4.48**	**0.103**
	Improvement	*2.1 (3.0%)*		*2.81 (53.4%)*		*33.9 (52.6%)*		*2.54 (42.4%)*		*73.9 (362.3%)*		*1.13 (17.1%)*	
Top-5	C&W$^*_{9\times30}$	75.8	2370	7.76	0.083	29.34	2425	7.94	0.086	0.7	2553	8.37	0.094
	AD$_{9\times30}$	**96.1**	1060	3.58	0.056	**80.68**	1568	5.13	0.070	**49.8**	2215	7.07	0.087
	C&W$^*_{9\times1000}$	100	437	1.59	0.044	100	600	2.16	0.058	100	779	2.77	0.074
	AD$_{9\times1000}$	100	**285**	**1.09**	**0.034**	100	**359**	**1.35**	**0.043**	100	**456**	**1.68**	**0.055**
Top-2	C&W$^*_{9\times30}$	99.9	1002	3.40	0.037	99.36	1504	4.95	0.050	97.9	2007	6.52	0.065
	AD$_{9\times30}$	99.9	308	1.12	0.028	99.5	561	1.94	0.037	98.4	873	2.92	0.049
	C&W$^*_{9\times1000}$	100	185	0.72	0.025	100	241	0.91	0.033	100	303	1.12	0.042
	AD$_{9\times1000}$	100	**137**	**0.56**	**0.022**	100	**174**	**0.70**	**0.028**	100	**220**	**0.85**	**0.035**
Top-1	C&W$_{9\times30}$	100	209.7	0.777	0.022	99.92	354.1	1.273	0.031	99.9	560.9	1.987	0.042
	AD$_{9\times30}$	100	140.9	0.542	0.018	99.9	184.6	0.696	0.025	99.9	238.6	0.880	0.032
	C&W$_{9\times1000}$	100	95.6	0.408	0.017	100	127.2	0.516	**0.023**	100	164.1	0.635	0.030
	AD$_{9\times1000}$	100	**81.3**	**0.380**	**0.016**	100	**109.6**	**0.472**	**0.023**	100	**143.9**	**0.579**	**0.029**
	FGSM	2.3	9299	24.1	0.063	0.46	9299	24.1	0.063	0	N.A.	N.A.	N.A.
	PGD$_{10}$	99.6	4691	14.1	0.063	88.1	4714	14.2	0.063	57.1	4748	14.3	0.063
	MIFGSM$_{10}$	100	5961	17.4	0.063	99.98	6082	17.6	0.063	99.9	6211	17.9	0.063

4.2 Are all Top-k targets equally difficult to attack?

Intuitively, we understand that they should not be equally difficult. We conduct some experiments to test this hypothesis. In particular, we test whether the label semantic knowledge can help identify the weak spots of different attack methods, and whether the proposed AD method can gain more in those weak spots. We test Top-5 using ResNet-50 [3]. Table. 3 summarizes the results. We observe that for the 9×30 search budget, attacks are more challenging if the Top-5 targets are selected from the least-like set in terms of the label semantic similarity (see Eqn. 9), or from the lowest-score set in terms of prediction scores on clean images.

4.3 Results for DenseNet-121

To investigate if the observations from ResNets hold for other DNNs, we also test DenseNet-121 (Huang et al., 2017) in ImageNet-1000. We test two settings: $k = 1, 5$ [4]. Overall, we obtain similar results. Table. 4 summarizes the results.

[3] More results on other k's and pretrained DNNs will be tested.

[4] Due to computation demand and limited computing resources, we will add results for other k's.

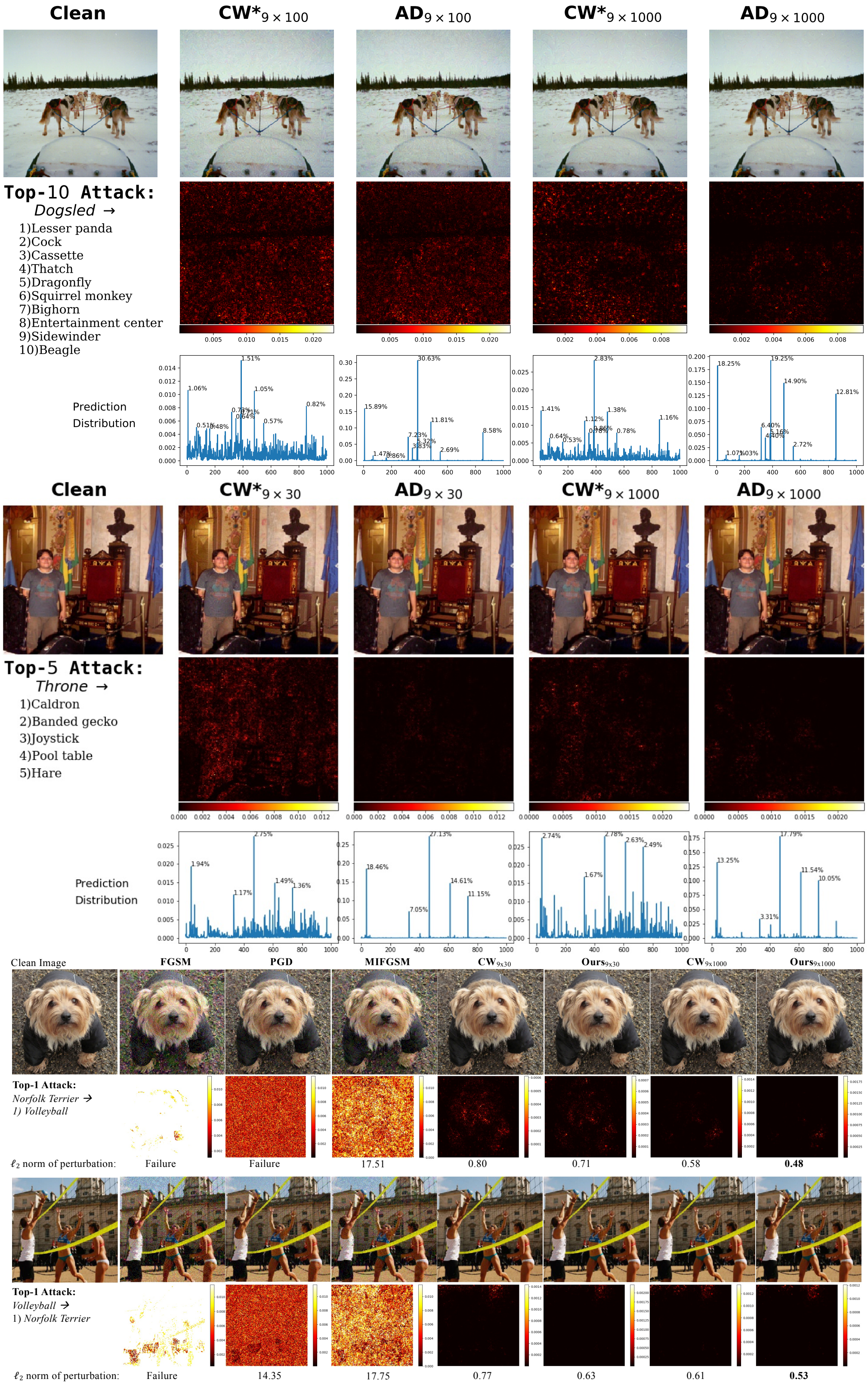

Figure 2: Learned adversarial examples for ordered Top-10 (top), Top-5 (middle) and Top-1 (bottom) attacks using ResNet-50 (He et al., 2016). The proposed AD method has smaller perturbation energies and "cleaner" (lower-entropy) prediction distributions. Note that for Top-10 attacks, the 9 × 30 search scheme does not work (see Table. 2).

Table 3: Results of ordered Top-5 targeted attacks with targets being selected based on (Top) label similarity, which uses 5 most-like labels and 5 least-like labels as targets respectively, and (Bottom) prediction score of clean image, which uses 5 highest-score labels and 5-lowest score labels. In both cases, GT labels are exclusive.

Protocol	Similarity	Method	ASR	ℓ_1	ℓ_2	ℓ_∞
Label similarity	Most like	C&W*$_{9\times30}$	80	1922	6.30	0.066
		AD$_{9\times30}$	**96.5**	1286	4.20	0.054
		C&W*$_{9\times1000}$	100	392	1.43	0.042
		AD$_{9\times1000}$	100	**277**	**1.05**	**0.035**
	Least like	C&W*$_{9\times30}$	27.1	2418	7.90	0.085
		AD$_{9\times30}$	**77.1**	1635	5.35	0.072
		C&W*$_{9\times1000}$	100	596	2.15	0.060
		AD$_{9\times1000}$	100	**370**	**1.39**	**0.045**
Prediction Score	Highest	C&W*$_{9\times30}$	93	1546	4.98	0.042
		AD$_{9\times30}$	**99.9**	1182	3.78	0.039
		C&W*$_{9\times1000}$	100	205	0.75	0.025
		AD$_{9\times1000}$	100	**170**	**0.65**	**0.023**
	Lowest	C&W*$_{9\times30}$	13.4	2231	7.30	0.082
		AD$_{9\times30}$	**68.6**	1791	5.86	0.077
		C&W*$_{9\times1000}$	100	621	2.25	0.064
		AD$_{9\times1000}$	100	**392**	**1.47**	**0.047**

Table 4: Results and comparisons using DenseNet-121 Huang et al. (2017) under the ordered Top-5 and Top-1 targeted attack protocol using randomly selected and ordered 5 targets (GT exclusive). For Top-1 attacks, we also compare with three state-of-the-art untargeted attack methods, FGSM (Goodfellow et al., 2015), PGD (Madry et al., 2018) and MIFGSM (Dong et al., 2018). 10 iterations are used for both PGD and MIFGSM.

Protocol	Method	Best Case				Average Case				Worst Case			
		ASR	ℓ_1	ℓ_2	ℓ_∞	ASR	ℓ_1	ℓ_2	ℓ_∞	ASR	ℓ_1	ℓ_2	ℓ_∞
Top-5	C&W*$_{9\times30}$	96.6	2161	7.09	0.071	73.68	2329	7.65	0.080	35.6	2530	8.28	0.088
	AD$_{9\times30}$	**97.7**	6413	2.14	0.043	**92.66**	1063	3.57	0.057	**83.3**	1636	5.35	0.072
	C&W*$_{9\times1000}$	100	392	1.42	0.040	100	527	1.89	0.052	100	669	2.37	0.065
	AD$_{9\times1000}$	100	**273**	**1.05**	**0.033**	100	**344**	**1.29**	**0.042**	100	**425**	**1.57**	**0.052**
Top-1	C&W$_{9\times30}$	99.9	188.6	0.694	0.019	99.9	279.4	1.008	0.028	99.9	396.5	1.404	0.037
	AD$_{9\times30}$	99.9	136.4	0.523	0.017	99.9	181.8	0.678	0.024	99.9	240.0	0.870	0.031
	C&W$_{9\times1000}$	100	98.5	0.415	**0.016**	100	132.3	0.528	**0.023**	100	174.8	0.657	**0.030**
	AD$_{9\times1000}$	100	**83.8**	**0.384**	**0.016**	100	**115.9**	**0.485**	**0.023**	100	**158.69**	**0.610**	**0.030**
	FGSM	6.4	9263	24.0	0.063	1.44	9270	24.0	0.063	0	N.A.	N.A.	N.A.
	PGD$_{10}$	100	4617	14.2	0.063	97.2	4716	14.2	0.063	87.6	4716	14.2	0.063
	MIFGSM$_{10}$	100	5979	17.6	0.063	100	6095	17.6	0.063	100	6218	17.9	0.063

5 Conclusions and Discussions

This paper proposes to extend the traditional Top-1 targeted attack setting to the ordered Top-k setting ($k \geq 1$) under the white-box attack protocol. The ordered Top-k targeted attacks can improve the robustness of attacks themselves. To our knowledge, it is the first work studying this ordered Top-k attacks. To learn the ordered Top-k attacks, we present a conceptually simple yet effective adversarial distillation framework motivated by network distillation. We also develop a modified C&W method as the strong baseline for the ordered Top-k targeted attacks. In experiments, the proposed method is tested in ImageNet-1000 using two popular DNNs, ResNet-50 and DenseNet-121, with consistently better results obtained. We investigate the effectiveness of label semantic knowledge in designing the adversarial distribution for distilling the ordered Top-k targeted attacks.

Discussions. We have shown that the proposed AD method is generally applicable to learn ordered Top-k attacks. But, we note that the two components of the AD framework are in their simplest forms in this paper, and need to be more thoroughly studied: designing more informative adversarial distributions to guide the optimization to learn adversarial examples better and faster, and investigating loss functions other than KL divergence such as the Jensen-Shannon (JS) divergence or the Earth-Mover distance. On the other hand, we observed that the proposed AD method is more effective when computation budget is limited (*e.g.*, using the 9×30 search scheme). This leads to the theoretically and computationally interesting question whether different attack methods all will work comparably well if the computation budget is not limited. Of course, in practice, we prefer more powerful ones when only limited computation budget is allowed. Furthermore, we observed that both the modified C&W method and the AD method largely do not work in learning Top-k ($k \geq 20$) attacks with the two search schema (9×30 and 9×1000). We are working on addressing the aforementioned issues to test the Top-k ($k \geq 20$) cases, thus providing a thorough empirical answer to the question: *how aggressive can adversarial attacks be?*

References

Anish Athalye and Ilya Sutskever. Synthesizing robust adversarial examples. *CoRR*, abs/1707.07397, 2017. URL `http://arxiv.org/abs/1707.07397`.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pp. 535–541, 2006. doi: 10.1145/1150402.1150464. URL `https://doi.org/10.1145/1150402.1150464`.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. *CoRR*, abs/1608.04644, 2016. URL `http://arxiv.org/abs/1608.04644`.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *AISec@CCS*, pp. 15–26. ACM, 2017.

Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. EAD: elastic-net attacks to deep neural networks via adversarial examples. In *AAAI*, pp. 10–17. AAAI Press, 2018.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, pp. 9185–9193. IEEE Computer Society, 2018.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL `http://arxiv.org/abs/1412.6572`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. URL `http://arxiv.org/abs/1503.02531`.

Kurt Hornik, Maxwell B. Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. doi: 10.1016/0893-6080(89)90020-8. URL `https://doi.org/10.1016/0893-6080(89)90020-8`.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems (NIPS)*, pp. 1106–1114, 2012.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *ICLR (Workshop)*. OpenReview.net, 2017.

Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*. OpenReview.net, 2018.

Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 427–436, 2015. URL `http://dx.doi.org/10.1109/CVPR.2015.7298640`.

Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy*, pp. 582–597. IEEE Computer Society, 2016.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *In EMNLP*, 2014.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing neural networks by penalizing confident output distributions. *CoRR*, abs/1701.06548, 2017. URL `http://arxiv.org/abs/1701.06548`.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Jiawei Su, Danilo Vasconcellos Vargas, and Sakurai Kouichi. One pixel attack for fooling deep neural networks. *CoRR*, abs/1710.08864, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015. URL `http://arxiv.org/abs/1512.00567`.

Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016. URL `http://arxiv.org/abs/1602.07261`.

Kaidi Xu, Sijia Liu, Pu Zhao, Pin-Yu Chen, Huan Zhang, Deniz Erdogmus, Yanzhi Wang, and Xue Lin. Structured adversarial attack: Towards general implementation and better interpretability. *CoRR*, abs/1808.01664, 2018.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530, 2016. URL `http://arxiv.org/abs/1611.03530`.